Data Structures and Algorithms (DT505G)

First Examination VT2020 (Exam #32)

Teacher:	Federico Pecora (ext. 3319)
Published:	2020-03-26 at 14:15
No. exercises:	6
Total points:	70 (42 required to pass)
No. pages:	6 (excluding this page)

- This is a **personalized exam**
- Compile your answers into a **Word** or **PDF** file named `32_submission.pdf`
- Submission instructions
  - Open the course Blackboard page, go to the Course material section
  - Click on the item entitled **Exam 2020-03-26 assignment**
  - Submit your Word/PDF file as an **attachment** to this assignment
- Submissions are due **at 18:15 today** (unless you are given another deadline by Funka)
- The teacher is available **from 15:00 to 16:30 to answer questions** at:
  - Zoom (browser): `https://oru-se.zoom.us/j/586404018`
  - Telephone: `+46844682488` or `+46850520017` (meeting ID: 586404018)
  - Mobile: `+46844682488,,586404018#` or `+46850520017,,586404018#`
  - If you are put on hold, please wait, as this means there is a queue
- If further doubts arise, make reasonable assumptions and **write them down**

Exercise 1 (10 points)

Given the array $A = \langle 7, 16, 8, 10, 6, 5, 4, 15, 14, 11 \rangle$, show how the Quicksort and Insertion sort algorithms work. Step through each algorithm, illustrating how it modifies the input array A. State the worst- and best-case computational complexity of the two algorithms in terms of the size $|A|$ of the input array, and explain why.

Exercise 2 (12 points)

Construct a Binary Search Tree (BST) from the array of integers A in the previous exercise. Do this by inserting each integer into the tree, one by one, starting from the left (element '7'). Draw the resulting tree and state its height. Is the tree balanced?

Remove the third element of A (element '8') from the BST, explaining the operations that this procedure performs. Now, insert the element you removed back into the BST, and draw the resulting tree. Is the resulting BST different from the BST before the element was removed? If so, will this be the case for any element that is removed and then re-inserted?

Illustrate the algorithm for finding the successor of an element in a BST. Show how this works, step by step, for the fourth element of A (element '10'). In general, how many key comparisons are performed by the algorithm for finding the successor of an element in a BST? What is the computational complexity of the algorithm?

Construct a new BST using the sorted array you obtained in Exercise 1, again by inserting the elements of the array into the tree one by one. What is the height of this BST, and is it balanced?

Exercise 3 (12 points)

Given a tree, a path from a node v is a sequence of nodes from node v to a leaf node. The cost of a path is the sum of the keys of the nodes along the path. The *maximum cost of a tree* is the highest cost of any path from the root node. For example, the maximum cost of the tree below is 61, as this is the cost of the most expensive path from the root of the tree to a leaf.

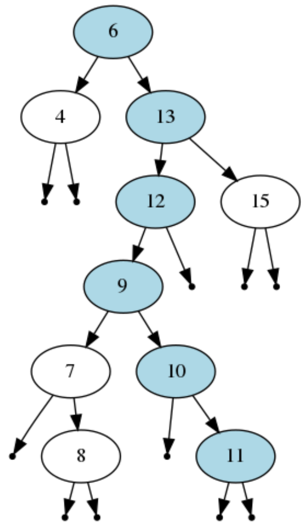


Figure 1: Nodes along theh path with maximum cost are highlighted in blue.

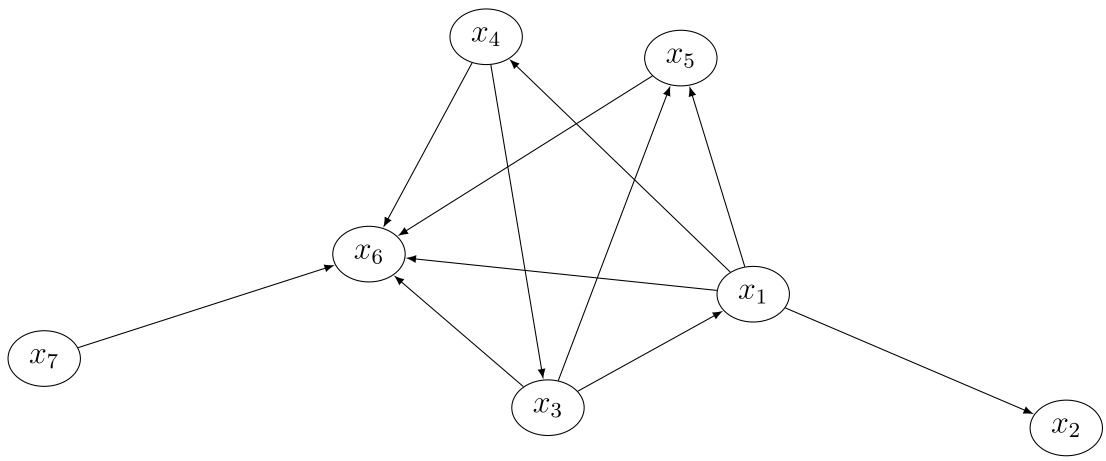
Write an algorithm that returns the maximum cost of a tree. Keep in mind that the maximum cost of a path from node v is equal to the cost of node v plus the maximum of the costs of the paths from its children. State the computational complexity of the algorithm, and demonstrate how your algorithm works with the first BST you built in Exercise 2.

Exercise 4 (12 points)

Given a directed, unweighted graph $G = (V, E)$, describe an algorithm to solve each of the following problems:

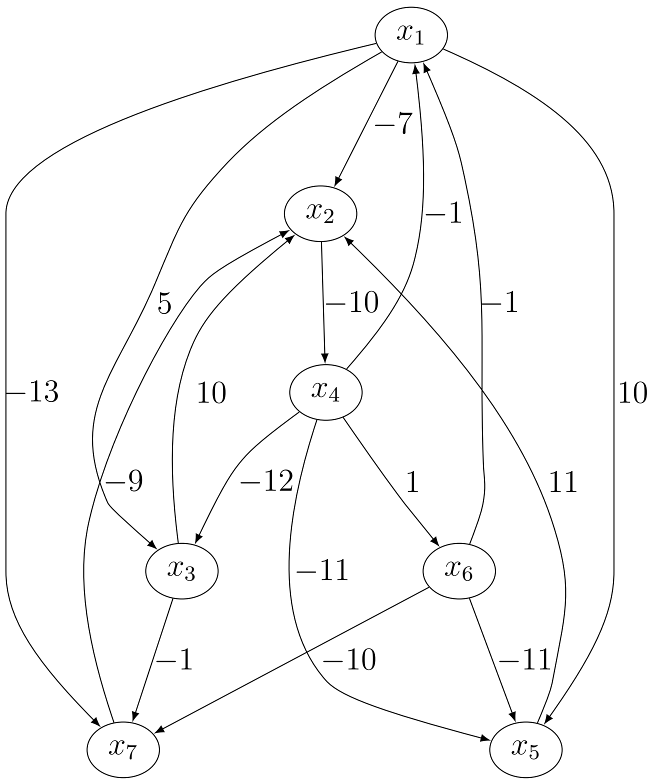
- Return whether the graph is connected.
- Return whether the graph is a tree.
- Find all the strongly connected components of the graph.
- Find the shortest path from node x_1 to every other node.

Show all the steps performed by each algorithm, using as input the graph G given below. Also, state and explain the computational complexity required to solve each problem in terms of the number of nodes $|V|$ and number of edges $|E|$ of the input graph.



Exercise 5 (12 points)

Given the following directed graph $G = (V, E)$ with weight funtion $w : E \mapsto \mathbb{Z} \setminus \{0\}$ as specified on the edges of the graph, illustrate an algorithm for finding a shortest path from vertex x_1 to every other vertex in the graph. Show how, at each step, the algorithm updates the distance estimate $d[x_i]$ of every vertex $x_i \in V$. State the computational complexity of the algorithm in terms of the number of nodes $|V|$ and number of edges $|E|$ of the input graph.



Exercise 6 (12 points)

State whether each of the following 6 statements on the asymptotic behavior of the function f is true or false.

1.	$f(n) = 2n^4 + n^2 + 2,$	$g(n) = 8^n,$	$f(n) \in \Theta(g(n))$
2.	$f(n) = 3n^4,$	$g(n) = 8n\log(n),$	$f(n) \in o(g(n))$
3.	$f(n) = 2n\log(n),$	$g(n) = 8n\log(n),$	$f(n) \in O(g(n))$
4.	$f(n) = 6^n,$	$g(n) = 2n^2 - n + 4,$	$f(n) \in o(g(n))$
5.	$f(n) = n^4 + 2n - 1,$	$g(n) = 3n^4 + 2n^3,$	$f(n) \in \Omega(g(n))$
6.	$f(n) = 8^n,$	$g(n) = 9^n,$	$f(n) \in \omega(g(n))$